



# 日 本 国 特 許 庁

PATENT OFFICE  
JAPANESE GOVERNMENT



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

1998年12月15日

出 願 番 号

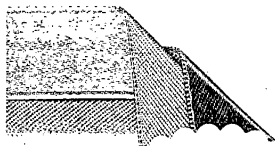
Application Number:

平成10年特許願第355757号

出 願 人

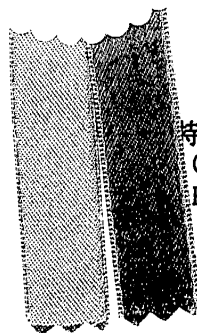
Applicant (s):

インターナショナル・ビジネス・マシーンズ・コーポレイション



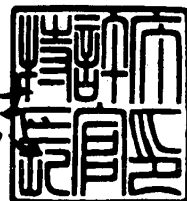
CERTIFIED COPY OF  
PRIORITY DOCUMENT

1999年 3月26日



特許庁長官  
Commissioner,  
Patent Office

山 佐 建 夫



出証番号 出証特平11-3017787

【書類名】 特許願

【整理番号】 JA998213

【あて先】 特許庁長官 殿

【国際特許分類】 G06T 15/00

【発明の名称】 半透明物体と不透明物体とが混在する複数の物体についての画像データをコンピュータ表示画面に表示する描画方法および描画装置

【請求項の数】 7

【発明者】

    【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ピー・エム株式会社 大和事業所内

    【氏名】 菰岡 晴夫

【発明者】

    【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ピー・エム株式会社 大和事業所内

    【氏名】 長谷川 一彦

【特許出願人】

    【識別番号】 390009531

    【住所又は居所】 アメリカ合衆国 1 0 5 0 4、ニューヨーク州アーモンク (番地なし)

    【氏名又は名称】 インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

    【識別番号】 100086243

    【弁理士】

    【氏名又は名称】 坂口 博

    【連絡先】 0 4 6 2 - 7 3 - 3 3 1 8、3 3 2 5、3 4 5 5

【選任した代理人】

    【識別番号】 100091568

【弁理士】

【氏名又は名称】 市位 嘉宏

【手数料の表示】

【予納台帳番号】 024154

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9304391

【包括委任状番号】 9304392

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】

半透明物体と不透明物体とが混在する複数の物体についての画像データをコンピュータ表示画面に表示する描画方法および描画装置

【特許請求の範囲】

【請求項 1】

奥行き方向についての情報をもつ不透明物体と半透明物体とが混在している複数の物体についての画像データを、記憶装置として更新可能な Z バッファを利用して、コンピュータ表示画面に表示する描画方法であって、

Z バッファの更新を行いながら、かつ、Z バッファによる隠面消去を行いながら、画像データのうちの不透明な物体だけを描画するステップと、

Z バッファの更新を行わないで、かつ、Z バッファによる隠面消去を行いながら、画像データのうちの半透明な物体だけを描画するステップと、

Z バッファの更新を行いながら、かつ、Z バッファによる隠面消去を行いながら、画像データのうちの半透明な物体だけを描画するステップと、  
を有する方法。

【請求項 2】

半透明物体を描画するステップが  $\alpha$  ブレンドによって実行される、  
請求項 1 記載の方法。

【請求項 3】

奥行き方向についての情報をもつ不透明物体と半透明物体とが混在している複数の物体についての画像データを、コンピュータ表示画面に表示することが可能な描画装置であって、

コンピュータ表示画面のドット毎に対応して設けられていて、奥行き情報を記憶する更新可能な Z バッファと、

コンピュータ表示画面のドット毎に対応して設けられていて、その内容を表示画面に出力することができ、表示するためのデータを記憶する更新可能な、フレームバッファと、

画像データを受け取って、これから描画しようとする画像データの奥行き情報

と既にZバッファに記憶されている奥行き情報とを比較して、これから描画しようとする画像データの奥行き情報が既にZバッファに記憶されている奥行き情報よりも大きいかな否かを判断することができ、奥行き情報の更新を行いながら表示するためのデータを出力することと奥行き情報の更新を行わないで表示するためのデータを出力することとを選択することが可能であり、かつ、既にフレームバッファに記憶されている表示するためのデータを読み出して、その後に受け取られる画像データとの間でコンピュータ表示画面のドット毎にブレンド処理することが可能である、レンダリングエンジンとを有する、コンピュータ表示画面に表示する描画装置。

【請求項4】

さらに、

フレームバッファから出力される表示するためのデータを、表示画面に表示することができる表示装置とを有する、

請求項3記載のコンピュータ表示画面に表示する描画装置。

【請求項5】

半透明物体についてのブレンド処理が $\alpha$ ブレンドによって実行される、

請求項3記載のコンピュータ表示画面に表示する描画装置。

【請求項6】

請求項1又は請求項2の描画方法を実行することが可能なラスタ・スキャン表示装置。

【請求項7】

請求項3乃至請求項5の何れかに記載の描画装置を有するラスタ・スキャン表示装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は、一般的には、コンピュータ・グラフィックス (CG: Computer Graphics) に関し、より具体的には、半透明な物体を表示するための描画順序の工夫に関するものである。

## 【0002】

## 【従来の技術】

CAD (Computer Aided Design)の分野においては、サーフェスやソリッドによって表現された3次元物体(object)又は3次元対象物をモデリングすることが一般的となってきた。このモデルとして表現されている画像(image)データを、画面(screen)上に写実的で現実感のある物体をつくりあげることは、レンダリングと総称されている。CGによる3次元表示は、モデリングとレンダリングが主なプロセスである。

## 【0003】

このレンダリングの過程においては、物体の表面の質感(素材感)をまるで写真と同じように現実に近い形で再現させるため、微かな陰(shade)や色の変化、色合い、反射、輝き、屈折、光源による影(shadow)のでき具合、周囲からの光の回り込み、映り込み、透明感、不透明感などが考慮されて、シェーディング(陰影づけ)が行われることも一般的になってきている。

## 【0004】

また、レンダリングの過程においては、各3次元物体のモデルは、最終的には3次元の物体空間から2次元の画像空間の投影面である画面(screen)へ投影される。この際、「奥行き方向」(「Z方向」とも呼ばれる)の情報(「奥行き情報」)に基づいて、視点(view point)又は投影中心からの透視投影(中心投影)によって、又は、無限遠点(infinite)からの平行な視線による平行投影によって、二次元領域であるコンピュータ表示画面(投影面)に塗り重ねるように投影されて表示される。

## 【0005】

塗り重ねるように投影されて表示されることは、具体的意味には、(1)表示バッファとして用意された記憶領域に表示される物体の画像データを書き込む(W: Write)ことのみによって、又は、(2)まずバッファなどとして用意された記憶領域に記憶されている画像データを読み出し(R: Read)、その読み出された画像データを加工し(M: Modify)、その加工された画像データを書き込む(W: Write)、ことが繰り返されることによるRMWサイクルによって、実行される。

すなわち、これが一般的な「描画する(draw)」ためのサイクル形態となる。

【0006】

このような投影をするにあたって、最も単純に隠線(hidden line)消去(removal)や隠面(hidden surface)消去(以下、隠面消去と略す)をすることができる手法であって、かつ、最も広く利用されている手法として、Zバッファ(Z-buffer)・アルゴリズム(Zバッファ法)という隠面消去アルゴリズムが周知である。個別にカラーや輝度の割当てが可能な表示面の最小単位である画素(ピクセル)を考えてみると、Zバッファ法ではその名が示すとおり、物体のZ値(視点又は投影中心等からの距離又は奥行き情報)を記憶するためのメモリ(デプスメモリ)を画素ごとに必要とし、全体として画素数分の大きさを持つ「Zバッファ」というメモリ配列を必要とする。

【0007】

Zバッファ法を用いながら不透明(opaque)色のみを用いて塗り重ねていく表示を進めていけば、最終結果としては完全な隠線消去や隠面消去を得ることができる。しかし、ユーザの視点から不透明な物体によって遮られてしまう奥の部分を確認することが困難になってしまう。つまり、Zバッファ法は不透明な物体の表示のために用いられるものと了解されている。一方、CADの分野では、半透明(semitransparent)に物体を表示することにより、表面のみならず内部も同時に確認できる表示方法も望まれている。

【0008】

半透明で表示する手法のひとつに、ブレンド(ブレンディング)を用いた手法がある。この手法では、描画すべき物体の色(src\_color)と、既に描画されている物体の色(dest\_color)とに対して係数A, Bを使い、新たな色(color)を、 $color = A \times src\_color + B \times dest\_color$ によって求める。より具体的には、既に描画されている物体の色(dest\_color)を読み出し(R)、ある物体の色(src\_color)によって加工し(M)、新たな色(color)を書き込む(W)ことによって実現される。つまり、読み出された物体の色(dest\_color)が、新たな色(color)に置き換わって表示されることになる。ここで

の新たな色 (color) への加工は、係数A, Bの乗算によって計算される。

【0009】

特に半透明係数 (ブレンディング・レートと呼ばれることもある)  $\alpha$  を用いる方法 ( $\alpha$  ブレンド) として以下の計算式 (1) が一般によく使用されている。

【0010】

$$\text{color} = \alpha \times \text{src\_color} + (1 - \alpha) \times \text{dest\_color} \dots\dots\dots (1)$$

【0011】

計算式 (1) を使って新たな色を計算し描画することを、以下「ブレンドする」、「ブレンド処理する」、「ブレンディング処理する」と表記する。ブレンドすると、同一点において、既に描かれていた物体よりも後から描いた物体の方が視点により近く見えるという特性がある。

【0012】

従ってこの手法で正確に半透明表示を行おうとすると、優先法 (method in list priority) を利用する必要に迫られる。すなわち、物体をユーザの視点からの距離によって (すなわち奥行き方向の深さによって) ソート (sort) をすることによって描画すべき物体の優先順位 (priority) を決めて、その優先順位リスト (list priority) を作成した上で、遠い方から順に描画する必要に迫られる。

【0013】

もしソートによって優先順位を決めないでこの手法を適用して描画した場合、物体同士の遠近関係が実際の位置関係とは異なって表示されてしまう場合が生じて問題となる。これが、物体同士の位置関係によらず、後から書いた物体が近くに見えるという現象であり、視点より近い物体が先に描画され、視点より遠い物体が後からブレンドされる場合に起きる問題である。従ってソートされていない以上、望んでいる通りの順序で描画が行われるとは限らず、望んだ通りの適切な強調表示が保証されないのである。

【0014】

このように、ユーザに厳密な遠近感を保証するという意味において高品質さを追求すると、ソートを厳密に行って半透明表示を行う必要が生じる。しかし、ソ



ートすべき対象となる物体の数が大きければ大きいほど、そのソート処理に要する時間が加速度的に多く必要となってくる。このような事態は、グラフィック全体に要する処理速度が著しく低下してしまい、大きな問題となる。概算してみると分かるのであるが、 $nP_n$  ( $n$ がソートすべき対象となる物体の数) という順列についてソートするためには、一般に、 $n \times \log(n)$  という処理数が必要となる。

#### 【0015】

CADなどのように対話的に（インタラクティブに又はリアルタイムに）データの編集処理を必要とするアプリケーションにとって、このようなソートの手順を採用することには問題がある。特に、複数の自由曲面を編集している場合、曲面を分割したファセット一つが1ピクセルで表現されている場合においては、全画素毎にソートが必要となってくる、一般的な $1280 \times 1024$ ピクセルの画面を想定した場合には、全体的な処理速度の低下は、もはや実用に耐えられるものではなくなる。

#### 【0016】

##### 【発明が解決しようとする課題】

すなわち、厳密さを追求して完全なソートをしなくても、ユーザにとって視覚上問題のない範囲内において簡易迅速に $\alpha$ ブレンディング処理ができるのであれば、それで十分である。一方で、対話的に（インタラクティブに）データの編集処理をするアプリケーションにおいては処理速度が重要となってくるため、ソートを省ける高速処理が実現できればきわめて好ましい。

#### 【0017】

##### 【課題を解決するための手段】

本発明を実現できる方法として、奥行き方向についての情報をもつ不透明物体と半透明物体とが混在している複数の物体についての画像データを、記憶装置として更新可能なZバッファを利用して、コンピュータ表示画面に表示する描画方法であって、Zバッファの更新を行いながら、かつ、Zバッファによる隠面消去を行いながら、画像データのうちの不透明な物体だけを描画するステップと、Zバッファの更新を行わないで、かつ、Zバッファによる隠面消去を行いながら、

画像データのうちの半透明な物体だけを描画するステップと、Zバッファの更新を行いながら、かつ、Zバッファによる隠面消去を行いながら、画像データのうちの半透明な物体だけを描画するステップと、を有する3つのパスを有するマルチパス半透明処理方法が提供される。

## 【0018】

また、本発明のマルチパス方法を実現できるハードウェア資源として、奥行き方向についての情報をもつ不透明物体と半透明物体とが混在している複数の物体についての画像データを、コンピュータ表示画面に表示することが可能な描画装置であって、コンピュータ表示画面のドット毎に対応して設けられていて、奥行き情報を記憶する更新可能なZバッファと、コンピュータ表示画面のドット毎に対応して設けられていて、その内容を表示画面に出力することができ、表示するためのデータを記憶する更新可能な、フレームバッファと、画像データを受け取って、これから描画しようとする画像データの奥行き情報と既にZバッファに記憶されている奥行き情報とを比較して、これから描画しようとする画像データの奥行き情報が既にZバッファに記憶されている奥行き情報よりも大きいかな否かを判断することができ、奥行き情報の更新を行いながら表示するためのデータを出力することと奥行き情報の更新を行わないで表示するためのデータを出力することとを選択することが可能であり、かつ、既にフレームバッファに記憶されている表示するためのデータを読み出して、その後に受け取られる画像データとの間でコンピュータ表示画面のドット毎にブレンド処理することが可能である、レンダリングエンジンとを有する、コンピュータ表示画面に表示する描画装置、が提供される。

## 【0019】

## 【発明の実施の形態】

図1は、コンピュータグラフィックスの処理をするためのハードウェア構成を示す。ここでは、近年最も普及しているものとして、陰極線管(CRT: Cathode-Ray Tube)でデータのリフレッシュが行われるラスタスキャンCRT(raster monitor CRT)10について説明するが、当業者にとって、液晶表示装置(LCD: Liquid Crystal Display)等の他の表示装置に置き換えて考えることはきわめ

て容易である。

#### 【0020】

画面（スクリーン）12に表示すべき図形情報をデジタル化し、そのデジタル化したデータをフレームバッファメモリ14のなかにドット単位で一時記憶させておく。フレームバッファ内でのドット単位の情報は、赤、緑、青のそれぞれの輝度の情報（8ビット分あれば $2^8$ の状態表現が可能）又はそれに相当する情報から成り立っている。ピクセルとフレームバッファメモリ14とは1対1の対応関係にしておくため、1ピクセルは1ドットに対応している。

#### 【0021】

フレームバッファメモリ内のデジタル化されたデータは、CRTコントローラ16に送られて映像信号に変換され、CRT18において電子ビームが振られて走査線上を移動されて、画面（スクリーン）12に所望の図形情報が表示される。

#### 【0022】

図2は、フレームバッファに送られるべき図形情報を得るまでの一般的なグラフィックス処理の流れを示す。物体又は対象物の幾何データ20がポリゴンに分割されてジオメトリエンジン22へ送られる。ジオメトリエンジン22では、ポリゴンの頂点の座標変換、クリッピングや光源、物体の材質情報などから頂点(vertex)の輝度を計算する。レンダリングエンジン24では、この頂点データと輝度情報をもとに各ピクセルの色を補間により計算し、Zバッファ26により隠面処理してフレームバッファに書き込まれる。

#### 【0023】

ところで、Zバッファ、隠面処理、隠線処理、シェーディングなどは、CPUとは独立したグラフィックス専用のグラフィックスプロセッサをもつ高速グラフィックス・アクセラレータ(accelerator)によって処理することもできる。

#### 【0024】

図3は、グラフィックス処理を実現するコンピュータのアーキテクチャの種類を示す図である。タイプAは、すべての処理をWSのCPUで処理して表示用のフレームバッファ14だけを持つものである。タイプBは、ジオメトリエンジン

22とレンダリングエンジン24とをハードウェア化してパイプライン構造にしたものである。タイプCは、レンダリングエンジン24のみをハードウェア化して、ジオメトリエンジン22で行う処理はCPU側においてソフトウェアで行うものである。

## 【0025】

大量なデータを伴う複雑な処理にはタイプBやタイプCが用いられる。本発明における描画方法は、隠面消去のための比較と更新が独立に制御することができるZバッファ（ソフトウェアで制御できる擬似的なものでも可）及び $\alpha$ ブレンドの機能を持つフレームバッファを備えていれば、タイプA、タイプB、タイプCの何れかのタイプのディスプレイプロセッサを利用して実行することが可能である。特に、ハードウェアによる隠面消去及び $\alpha$ ブレンドの機能をもつタイプB及びタイプCに対して有益である。

## 【0026】

Zバッファ法は優先法的一种でありながら、その利点として、いったん全ての物体の描画が終わったあとでも、次々と新たな物体をつけ加えて描画できるという性質がある。Zバッファ・アルゴリズムは、一度に1ずつ物体を（他の物体とは無関係に）描いていくので、その前又はその後ろに描かれる物体に関する情報は保持しない。従って、コンピュータ内部で確保しなければならないメモリの記憶域も非常に少なく済む。

## 【0027】

もっとも、Zバッファ法には、いったん描いてしまった物体は消去することはできないという性質もある。これは、Zバッファ法ではモデル空間に他の物体が存在するか否かは隠面消去の手順に全く影響しないからである。すなわち、処理の途中のある時点においては「隠れて見えないはずの」物体も描画している場合がある。これは、処理のときに必要となるのは新たに描画しようとする物体のデータとZバッファだけである、ということに起因している。

## 【0028】

Zバッファ法の処理においては、順次更新を行いながら次の描画のためにその更新された値を利用していくが、更新を行わないのであれば（すなわち更新を止

めておけば) そのままの値を維持して利用することもできる。

【0029】

そこで、本発明では、Zバッファ法にある新たな物体を後からつけ加えて描画できるという性質を、Zバッファの更新のタイミングを考えつつ $\alpha$ ブレンドを用いた半透明表示の手法に巧みに組み合わせて利用する。このことによって、ソートの手順という厳密ではあったものの処理には時間を要していた手順を省略することに成功したものである。

【0030】

すなわち、距離によるソートを行ってから正確な半透明物体の表示をするという考え方は捨てたが、描画順によらず実際の位置関係に近い半透明表示を行うために重要であると考えられる以下の条件1乃至3に着目してみる。これらは、ユーザにとって視覚上問題にならないように配慮された条件である。

【0031】

条件1： 視点から一番近くにある物体は、半透明、不透明にかかわらず、一番最後にブレンドまたは描画されるべきである。これは半透明、不透明にかかわらず、最後に $\alpha$ ブレンド又は描画されたものが、それ以前に $\alpha$ ブレンド又は描画されたものよりも近くに見えるという特性による。

【0032】

条件2： 不透明物体より遠くにある物体は、半透明、不透明にかかわらず表示されない。これは、不透明物体が後方の物体を遮るという性質に忠実に従っているものである。

【0033】

条件3： 半透明物体より遠くにある物体は、条件2によって表示されない場合を除いて、半透明、不透明にかかわらず $\alpha$ ブレンドされて表示される。

【0034】

これらの条件1乃至3を満たすことは、必要とされていたソートの手順に代えて、以下のような3段階に分けたマルチパス処理（半透明物体の処理に特徴があるので「マルチパス半透明処理」とも称する）によって達成できる。

【0035】

パス1： 不透明な物体だけをZバッファ法による隠面消去を行いながら描画していく。この場合にZバッファの更新も行う。

【0036】

パス2： 半透明な物体だけをZバッファ法による隠面消去を行いながら計算式(1)によってブレンドしていく。この場合にZバッファの更新は行わないで  
おく。

【0037】

パス3： 半透明な物体だけをZバッファ法による隠面消去を行いながら計算式(1)によってブレンドしていく。この場合にZバッファの更新も行う。

【0038】

従来技術における「単一パス」では、Zバッファ法による隠面消去を行わないで、かつ、データのソートを行わないで、データを与えられた順番に従って描画していたのであり、マルチパスでは、パス数としては3パスで描画する。

【0039】

ここで、パス1乃至3の各パスの作用をさらに詳細に説明していく。

【0040】

パス1においては、不透明な物体だけ进行处理する。このとき通常の不透明な物体の処理と同じくZバッファによる隠面消去を行い、Zバッファを更新する。計算式(1)による $\alpha$ ブレンドは行わない。結果として視点から一番近い不透明な物体が表示されることになり、Zバッファの値はそれに対応した値となる。パス2やパス3において不透明な物体は描画されないので、不透明な物体に関しては条件2を満たすことになる。

【0041】

パス2においては、半透明な物体だけ进行处理する。Zバッファによる隠面消去を行いながら計算式(1)による $\alpha$ ブレンドを行い、Zバッファの更新は行わない。従って、Zバッファによる隠面消去により、パス1で描画された不透明物体より近くにある物体だけについて $\alpha$ ブレンドを行うのである。Zバッファがパス1において不透明な物体まで更新されているので、不透明な物体より遠くにある半透明な物体は描画されることはなく、半透明な物体に関しても条件2を満たす

ことになる。このパス2ではZバッファの更新は行わないため、パス1において描画した不透明な物体より近くにある半透明な物体は常にブレンドされることになり、条件3を満たしている。

## 【0042】

パス2終了時点のZバッファの各点の値は、パス1終了時点と同じ値が保たれている。すなわち、この時点でもZバッファは各点において視点より一番近い不透明物体までの距離を保持している。

## 【0043】

パス3においては、再度、半透明な物体だけ进行处理する。Zバッファによる隠面消去を行いながら計算式(1)による $\alpha$ ブレンドを行い、Zバッファの更新も行う。従って、パス2とパス3との違いはZバッファの更新を行うという点である。パス2においては、Zバッファを更新しないために二つの半透明な物体はそれらの位置関係によらずとも両方 $\alpha$ ブレンドされた。

## 【0044】

一方で、パス3においては、Zバッファを更新するため、近い物体を先に処理すると、後から処理する物体は $\alpha$ ブレンドされないことになる。パス3での処理中は、Zバッファは各点での最後に $\alpha$ ブレンド又は描画された物体の視点からの距離を保持する。従って、最後に $\alpha$ ブレンドされることになるのは、視点に最も近い物体ということになり、条件1を満たしている。また、不透明な物体と視点との間に半透明な物体が無い場合には、パス1によって描画された物体が最後に描かれることになって、この場合でも条件1を満たしている。

## 【0045】

ここで注意されたいのは、視点に最も近い(不透明、半透明を問わない)物体とその他の(不透明、半透明)物体との間での相対的な位置関係の見え方は何ら損なわれることはないが、これらの間に挟まれることになる半透明物体の位置関係の見え方は必ずしも保証されないということである。これは正確なソートの手順を省略しているからである。

## 【0046】

最も近い物体ではない物体同士の位置関係は、アプリケーションによる描画順

に依存する。例えば、対話的（インタラクティブ）処理の典型であるCADにおいては、物体を回転させて反対方向から見るような操作がよく行われる。このため、視点から遠い順に物体のデータを定義し、その順番に描画することはソートを行わない限り不可能である。しかし、これらの物体の位置は、視点に最も近い（不透明、半透明を問わない）物体の位置に比べると、重要性が小さく十分妥協できるものと言える。

## 【0047】

ここで、RMWサイクルとの関連で、「描画する(draw)」ことをどのように実行できるかについて説明する。パス1における描画は、必ずしもR、M及びWの全てを実行する必要はなく、不透明物体の画像データによる置換としてWのみを実行すれば足りる。ただし、パス2における描画とパス3における描画では、現在の背景色（既に描画されている物体の色（dest\_color））と、これから描こうとする半透明物体の色（描画すべき物体の色（src\_color））とをブレンドする必要があるため、R、M及びWの全てを実行する必要がある。

## 【0048】

ここで、より具体的な例として、不透明物体A、半透明物体B、半透明物体C、といった3つの描画すべき対象がある場合を想定する。グラフィックス描画のためのアプリケーションプログラムが必ず、A、B、Cの順で描画した場合において本願発明の手法であるパス1乃至3が与える表示結果を考えてみることにする。

## 【0049】

ここでは記号の<を使って、視点からAがBより近くにあることを $A < B$ と表わすことにする。以下、A、B及びCの3つの物体が視点から見て重なり合った部分について考察する。

## 【0050】

$C < B < A$ の場合（視点からCが最も近く、Aが最も遠い場合）：

パス1ではAを描画する。

パス2ではBをブレンドしてからCをブレンドする。



パス3でもBをブレンドしてからCをブレンドする。

結果として、Cが最も近くもありその後ろにB、Aがあるように見える。

【0051】

図5は、 $C < B < A$ の場合（視点からCが最も近く、Aが最も遠い場合）の描画結果を（後に言及する図6の描画結果よりも）詳細に説明する図である。ここで物体Bに関係する部分については全てが同一方向に揃った斜線（右上から左下への斜線）によって表わされており、ここで物体Cに関係する部分については物体Bとは異なる方向であるが全て同一方向に揃った斜線（左上から右下への斜線）によって表わされている。また、言及する描画領域は、それぞれが単一種類の斜線部分だけではなく、これらが交差することによる組み合わせによっても特定される。より具体的には、右上から左下への斜線と左上から右下への斜線とが交差している、網目状に表わされた描画領域も存在する。

【0052】

描画領域a bはAとBとが重なることになる部分であり、Bの後方にAが見えるように描画される。描画領域b cはBとCとが重なることになる部分であり、Cの後方にBが見えるように描画される。描画領域a cはAとCとが重なることになる部分であり、Cの後方にAが見えるように描画される。さらには、描画領域a b cはA、B及びCの全てが重なる部分であり、Aの前方にBが見えてさらにBの前方にCが見えるように描画される。これらのうち描画領域b cと描画領域a b cとの区別は、物体Aと重なることになるかどうかによって、さらに網目状の密度が異なって表現されている。

【0053】

図6には、 $C < B < A$ の場合を含め、A、B、Cの3つの物体に生じ得る全ての位置関係を示している。なお、 $C < B < A$ 以外の場合については、図5に示したような描画結果の詳細な表現や説明については省略する。

【0054】

$B < C < A$ の場合（視点からBが最も近く、Aが最も遠い場合）：

パス1ではAを描画する。

パス2ではBをブレンドしてからCをブレンドする。

パス3ではBをブレンドするがCはブレンドされない。

結果として、Bが最も近くもありその後ろにC、Aがあるように見える。

【0055】

$B < A < C$  の場合（視点からBが最も近く、Cが最も遠い場合）：

パス1ではAを描画する。

パス2ではBのみブレンドされ、CはAに隠されてブレンドされない。

パス3ではBのみブレンドされ、CはAに隠されてブレンドされない。

結果として、Bの後ろにAがあるように見える。

【0056】

$C < A < B$  の場合（視点からCが最も近く、Bが最も遠い場合）：

パス1ではAを描画する。

パス2ではCのみブレンドされ、BはAに隠されてブレンドされない。

パス3ではCのみブレンドされ、BはAに隠されてブレンドされない。

結果として、Cの後ろにAがあるように見える。

【0057】

$A < B < C$  の場合（視点からAが最も近く、Cが最も遠い場合）：

パス1ではAを描画する。

パス2では何も描画されない。

パス3では何も描画されない。

結果として、Aのみが表示される。

【0058】

$A < C < B$  の場合（視点からAが最も近く、Bが最も遠い場合）：

パス1ではAを描画する。

パス2では何も描画されない。

パス3では何も描画されない。

結果として、Aのみが表示される。

【0059】

以上の例では、Aを最初に描画しているが、不透明な物体はパス1において処理され、半透明な物体はパス2およびパス3で処理されるため、アプリケーション

ンプログラムでの描画順によることなく、不透明な物体は常に半透明な物体よりも先に処理されることになる。従って、不透明物体が2番目、3番目に描画されるケースでも同じ結果が得られることになる。

## 【0060】

本発明の手法は、半透明表示を正確に行うためのものではないことは既に述べた。しかし、本発明によるマルチパスで $\alpha$ ブレンドを使用したものは、単一パスのみで $\alpha$ ブレンドを使用したものに比べて、遠近の位置関係が保たれるように表示されていることが確認されている。

## 【0061】

## 【発明の効果】

ここで、実際のアプリケーション上でのパフォーマンス測定結果を表1に示す。ソートを行った場合の半透明描画についての測定結果は、実用に耐えられるものではなく実装することが困難であるため、得られていない。従って、ソートをする部分の論理処理に必要な時間は、シミュレーションの結果を測定し、単一パスの半透明処理時間に加えることによって、予想時間を求めて比較の対象としている。対話的（インタラクティブ）処理においては採用できないものと判断される。

## 【0062】

## 【表1】

描画手法の違いによる100回繰り返し描画時間

	描画手法	ベンチマーク1	ベンチマーク2
[1]	不透明	8.3秒	64.6秒
[2]	単一パス半透明	8.3秒	63.9秒
[3]	ソート予想時間	272.8秒	767.0秒
[4]	ソート後半透明	281.1秒	830.9秒
[5]	マルチパス半透明	15.0秒	129.2秒

## 【0063】

より具体的には、[1]と[2]とにおいては現状のg r a P H I G Sを使用

して描画している。g r a P H I G S (g r a P H I G Sは、I B M C o r p . の米国およびその他の国における商標)は、3次元(3D) g r a p h i c s A P I (Application Programming Interface)であり、I S O標準(Standard)であるP H I G S (Programmer's Hierarchical Interactive Graphics System)をI B M社が実装(implement)したものである。[2]はソートを行わない単一パスのブレンドによる半透明処理であって、ここでは位置関係は保存されない。[3]においてはシミュレーションプログラムで測定してソートにかかる時間を予想している。[4]については[2] + [3]と仮定することで求めている。[5]は現状のg r a P H I G Sを本発明のマルチパス半透明処理ができるように改造して実装して描画したものである。

【0064】

[1]と[2]とでは単一(1回の)パスで処理が済むのに対して、本発明が適用されている[5]ではパス1乃至3の3回のパスを必要としているものの、描画時間は不透明の場合及び従来手法の2倍程度といった実用範囲内に抑えることができている。実データを使用した描画結果をユーザに評価させた結果においても、視覚上の位置関係を保った表示と対話的(インタラクティブ)処理可能な描画速度との両立という点で十分に満足が得られている。

【0065】

図7に示すような関係からわかるように、本発明の方法は、g r a p h i c s A P Iのみならず、アプリケーション・レベルにおいて実施することも可能である。すなわち、A P Iを介してグラフィックス処理は実行されることになるが、A P Iを介する前段としてのアプリケーション・レベルにおいても実施することもでき、当業者であれば本発明を様々な方法で適用することができる。

【図面の簡単な説明】

【図1】

コンピュータグラフィックスの処理をするためのハードウェア構成である、ラスタスキャンC R Tの構成を示す図である。

【図2】

フレームバッファに送られるべき図形情報を得るまでの一般的なグラフィック

処理の流れを示す図である。

【図 3】

グラフィックス処理を実現するコンピュータのアーキテクチャの種類を示す図である。

【図 4】

マルチパス処理（マルチパス半透明処理）を示す流れ図である。

【図 5】

不透明物体 A、半透明物体 B、半透明物体 C、という 3 つの描画すべき対象の位置関係が  $C < B < A$  の場合（視点から C が最も近く、A が最も遠い場合）の場合について、本発明の手法であるパス 1 乃至 3 を用いて A、B、C の順で描画した場合において、表示結果を詳細に説明する図である。

【図 6】

不透明物体 A、半透明物体 B、半透明物体 C、という 3 つの描画すべき対象に生じ得る全ての位置関係について、これらを描画すべき物体を本発明の手法であるパス 1 乃至 3 を用いて A、B、C の順で描画した場合において、表示結果を示す図である。

【図 7】

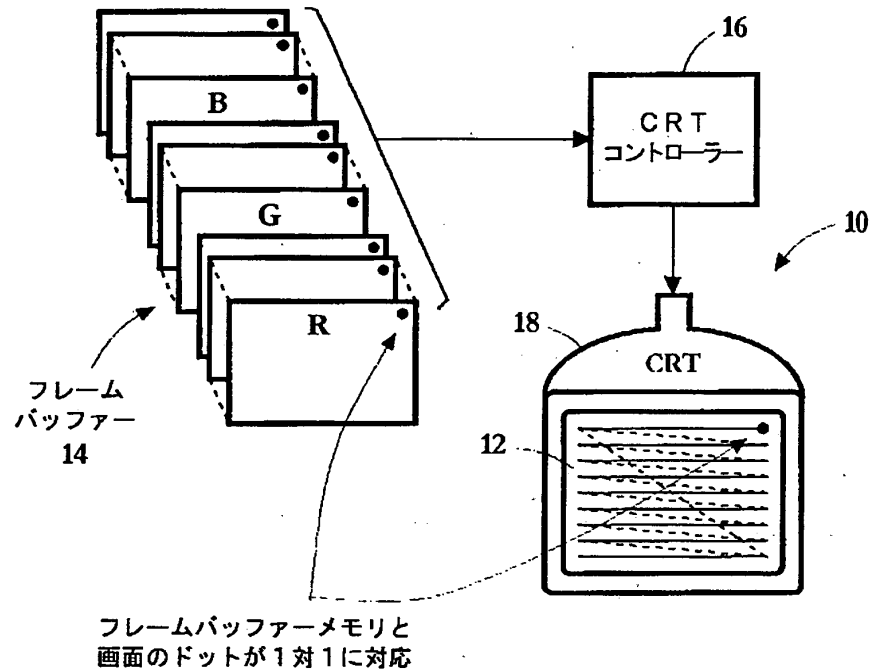
本発明の方法が、graphics API のみならず、アプリケーション・レベルにおいて実施することも可能であることを示す図である。

【符号の説明】

- 10 ラスタスキャン CRT
- 12 画面（スクリーン）
- 14 フレームバッファメモリ
- 16 CRT コントローラ
- 18 CRT
- 20 幾何データ
- 22 ジオメトリエンジン
- 24 レンダリングエンジン
- 26 Z バッファ

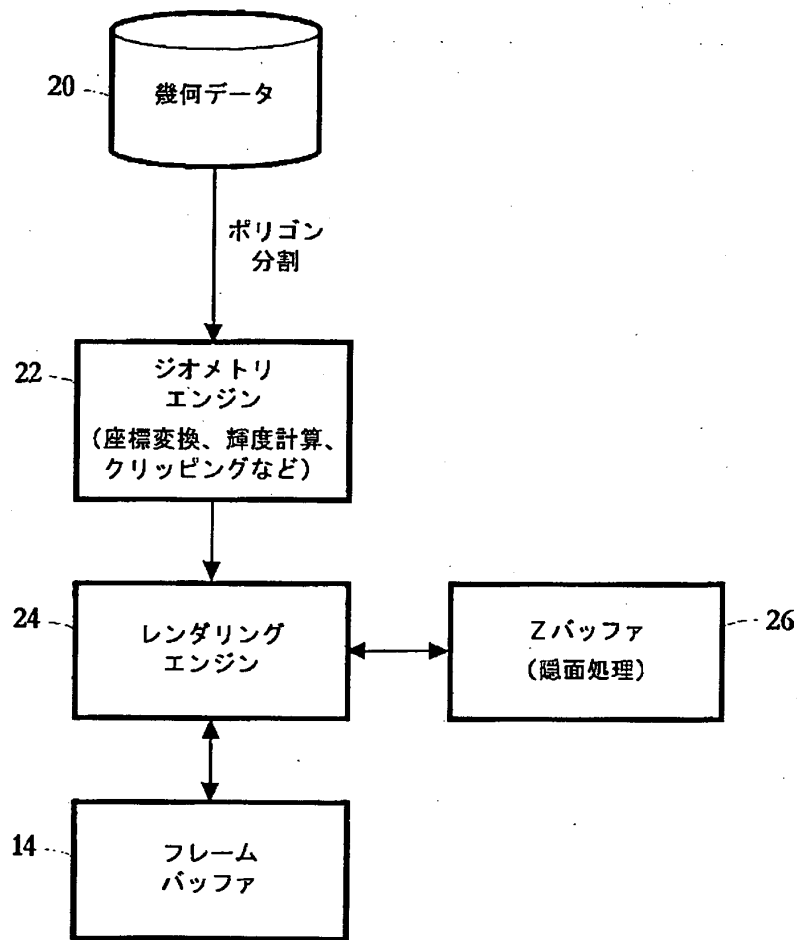
【書類名】 図面

【図 1】



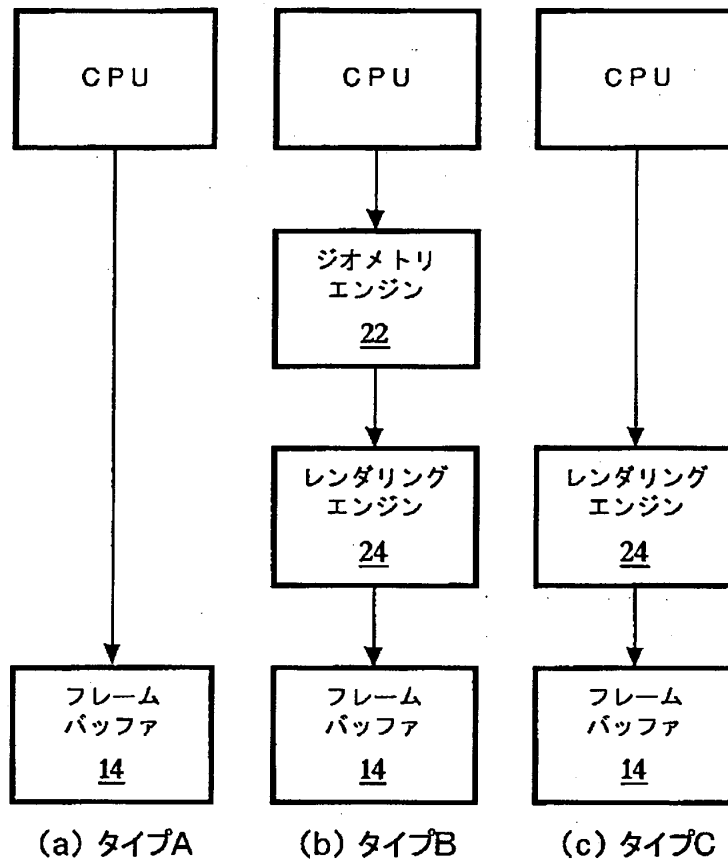
ラスタ・スキャンCRTの構造

【図2】



3次元グラフィックス処理

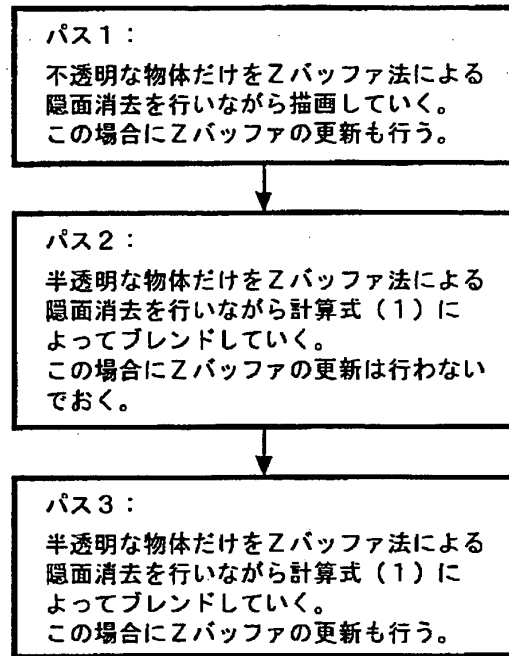
【図3】



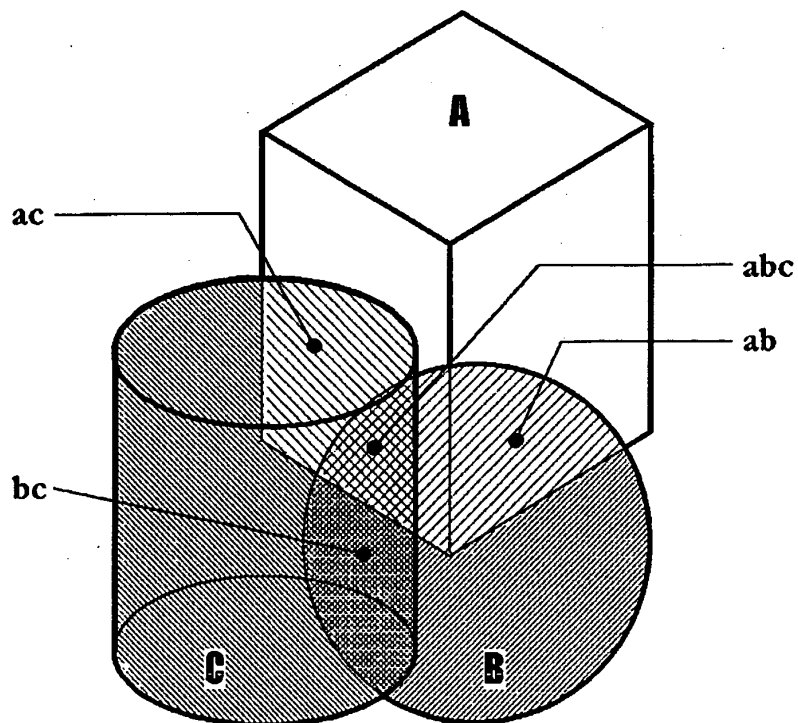
グラフィックスアーキテクチャ



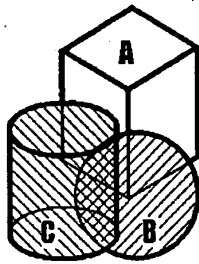
【図 4】



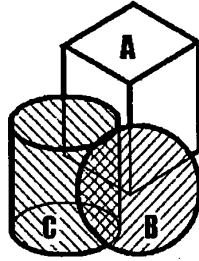
【図 5】



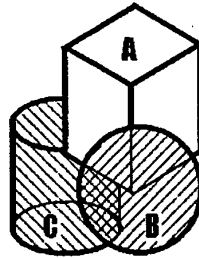
【図 6】



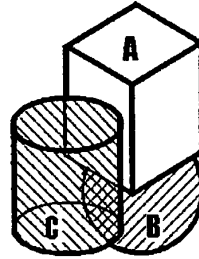
$$C < B < A$$



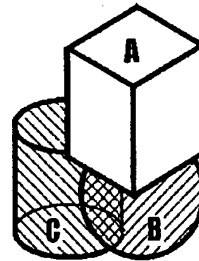
$$B < C < A$$



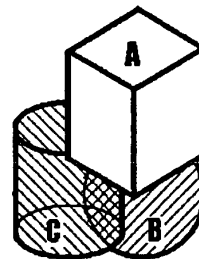
$$B < A < C$$



$$C < A < B$$

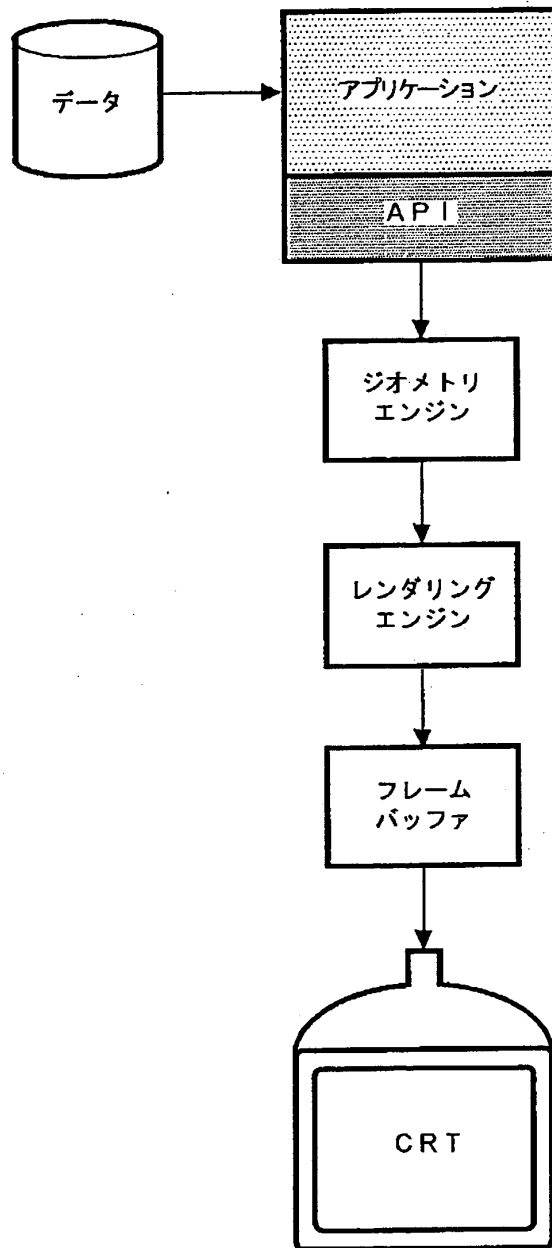


$$A < B < C$$



$$A < C < B$$

【図7】



【書類名】 要約書

【要約】

【課題】

マルチパス $\alpha$ ブレンドによるソートを行わないで済むような、半透明表示の方法を提供すること。

【解決手段】

描画順によらず実際の位置関係に近い半透明表示を行うために重要であると考えられる条件に着目して、まずパス1として、不透明な物体だけをZバッファによる陰面消去を行いながら描画する。Zバッファの更新も行う。次にパス2として、半透明な物体だけをZバッファによる陰面消去を行いながらブレンドする。この際にZバッファの更新は行わない。最後にパス3として、半透明な物体だけをZバッファによる陰面消去を行いながらブレンドする。この際にはZバッファの更新を行う。不透明表示、あるいは従来 of 簡易的な半透明表示が1回のパスで処理するのに対して、本手法は3回のパスを必要とするが、ハードウェアの追加を全く必要としないままで、描画時間は不透明の場合及び従来手法の2倍程度と実用範囲の中に押さえることができた。ユーザによる実データ使用の評価においても、視覚上の位置関係を保った表示とインタラクティブ処理可能な描画速度の両立という点で満足を得られている。

【選択図】 図4

【書類名】 職権訂正データ  
【訂正書類】 特許願

<認定情報・付加情報>

【提出日】 平成10年12月15日

【特許出願人】

【識別番号】 390009531

【住所又は居所】 アメリカ合衆国 10504、ニューヨーク州 アー  
モンク (番地なし)

【氏名又は名称】 インターナショナル・ビジネス・マシーンズ・コー  
ポレイション

【代理人】 申請人

【識別番号】 100086243

【住所又は居所】 神奈川県大和市下鶴間 1623 番地 14 日本アイ  
・ビー・エム株式会社 大和事業所内

【氏名又は名称】 坂口 博

【選任した代理人】

【識別番号】 100091568

【住所又は居所】 神奈川県大和市下鶴間 1623 番地 14 日本アイ  
・ビー・エム株式会社 大和事業所内

【氏名又は名称】 市位 嘉宏

出 願 人 履 歴 情 報

識別番号 [390009531]

1. 変更年月日 1990年10月24日

[変更理由] 新規登録

住 所 アメリカ合衆国10504、ニューヨーク州 アーモンク (番地なし)

氏 名 インターナショナル・ビジネス・マシーンズ・コーポレイション